

**VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky**

**Absolvování individuální odborné praxe  
Individual Professional Practice in the Company**

**2014**

**Slavomír Švancár**

## Zadání bakalářské práce

Student: **Slavomír Švancár**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ITA spol. s r.o. Ostrava
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
  - c) Zvolený postup řešení zadaných úkolů
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Peter Chovanec**


Konzultant bakalářské práce: Ing. Daniel Hajduk, Ph.D.

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## **Prehlásenie študenta**

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

Dňa: 7.5.2014

  
.....  
podpis študenta

## **Pod'akovanie**

Rád by som poďakoval firme ITA spol. s.r.o. za možnosť vykonať odbornú prax, konzultantovi Ing. Danielovi Hajdukovi, Ph.D za odbornú pomoc a prejavenu dôveru pri vykonávaní práce a zamestnancom za ochotu poradiť a vysvetliť všetky nejasnosti.

## **Prehlásenie zástupcu spolupracujúcej právnickej alebo fyzickej osoby**

„Súhlasím so zverejnením tejto bakalárskej práce podľa požiadaviek čl. 26, odst. 9  
Studijného a zkušebného rádu pro studium v bakalárských programech VŠB-TU Ostrava.“

Dňa: 7.5.2014

.....  
podpis zástupcu



## **Abstrakt**

Táto práca popisuje odbornú prax v spoločnosti ITA spol. s.r.o. V druhej kapitole je popísaná táto spoločnosť a jej hlavné aktivity. Ďalej sú popísané teoretické časti zadaných problémov počas praxe a ich riešenia. Za riešením nasleduje stručný popis použitých technológií MFC, OpenGL, GDI+ a knižnice ContourLib, ktorú vytvorila spoločnosť ITA. V závere sú zhrnuté využité teoretické poznatky nadobudnuté počas štúdia na vysokej škole a poznatky, ktoré mi pri riešení chýbali.

## **Kľúčové slová**

ITA, C++, MFC, OpenGL, GDI+, odborná prax

## **Abstract**

This thesis describes Professional practice in ITA, company with limited liability. This company and its main activities are described in second chapter. After that, there are described theoretical parts of assigned problems and their solutions. Solutions are followed by brief description of MFC, OpenGL, GDI+ Technologies, and ContourLib library created by ITA company. There are summarized theoretical knowledge acquired during college stury and knowledge missing during dealing with problems in the closing part.

## **Key words**

ITA, C++, MFC, OpenGL, GDI+, Professional practice

## Zoznam použitých skratiek

Skratka	Anglický význam	Slovenský význam
<b>API</b>	Application programming interface	Rozhranie pre programovanie aplikácií
<b>CAD</b>	Computer aided design	Počítačom podporovaný návrh
<b>GDI</b>	Graphics device interface	Rozhranie pre grafické zariadenia
<b>MFC</b>	Microsoft foundation class	
<b>MDI</b>	Multiple document interface	Rozhranie s viac dokumentami
<b>SDI</b>	Single document interface	Rozhranie s jedným dokumentom

# Obsah

1.	Úvod.....	1
2.	Profil spoločnosti .....	2
2.1	Prvé stretnutie s firmou .....	2
2.2	O spoločnosti.....	2
3.	Používané technológie .....	3
3.1	ContourLib .....	3
3.2	MFC .....	5
3.3	OpenGL.....	5
3.4	GDI+ .....	6
4.	Zadané úlohy a postup pri riešení .....	7
4.1	Projekt 1 .....	7
4.1.1	Teória .....	7
4.1.2	Riešenie problému .....	8
4.2	Projekt 2 .....	9
4.2.1	Teória .....	9
4.2.2	Riešenie problému .....	10
4.3	Projekt 3 .....	12
4.3.1	Valcovňa za tepla .....	12
4.3.2	Riešenie problému .....	13
5.	Teoretické a praktické poznatky .....	15
6.	Záver .....	16
	Použitá literatúra .....	17



## 1. Úvod

Rozhodol som sa vybrať si odbornú prax ako formu svojej bakalárskej práce, pretože som chcel ešte počas svojho štúdia nadobudnúť skúsenosti s prácou vo firme. Prax som absolvoval v spoločnosti ITA spol. s r.o. Ostrava. V druhej kapitole je firma bližšie predstavená a opísané hlavné zameranie firmy.

Počas svojej praxe som vytvoril pre firmu 3 komponenty použité v rôznych projektoch. Prvá časť bola najjednoduchšia. Jednalo sa len o prispôsobenie zobrazovania valca vo valcovni za studena tak, aby dokázala zobrazovať valce z dvoch materiálov. Druhá, komplexnejšia, časť, ktorej som sa venoval, bol komponent, ktorý má byť použitý vo vyvíjanom programe spoločnosti. Táto časť má slúžiť na vizualizáciu hodnôt zvlnenia oceľového pásu nameraných novým prístrojom, ktorý bol nainštalovaný na valcovňu za tepla. Tretia časť je použitá v riadiacom systéme valcovacích stolíc a slúži na zobrazovanie aktuálnej polohy valcovaného pásu vo valcovacích stolicích s pecnými navíjačkami.

Po popísaní častí práce, ktorým som sa venoval, nasleduje zhrnutie použitých technológií a ich krátky popis. Na záver sú popísané vedomosti nadobudnuté počas štúdia na vysokej škole, ktoré som použil pri vykonávaní mojej odbornej praxe ako aj znalosti, ktoré som si musel osvojiť samoštúdiom, prípadne s pomocou kolegov z práce.

## 2. Profil spoločnosti

### 2.1 Prvé stretnutie s firmou

O firme ITA spol. s.r.o.[1] som sa dozvedel prostredníctvom zoznamu firiem poskytovaných univerzitným systémom KatIS. Zaujala ma pozícia programátora v C++, a keďže tento programovací jazyk mi bol najbližší v rámci štúdia, rozhodol som sa kontaktovať túto firmu. Výberové konanie spočívalo vo vytvorení jednoduchej aplikácie na 3D vizualizáciu dát uložených vo formáte .xls (MS Excel), pričom bolo možné použiť akúkoľvek dostupnú technológiu. Na zobrazovanie som si vybral technológiu OpenGL, nakoľko bola OpenSource a vytvoril som jednoduchú konzolovú aplikáciu. Pár týždňov po odprezentovaní mojej aplikácie mi bola poskytnutá možnosť absolvovať odbornú prax v tejto firme.

### 2.2 O spoločnosti

ITA spol. s.r.o. je súkromná česká spoločnosť založená v roku 1991 výskumnými a vedeckými pracovníkmi Výskumného ústavu strojárského a metalurgického VÍTKOVICE. Hlavným zameraním firmy sú moderné technológie valcovania za tepla aj za studena. Firma dodáva know-how a programové riešenia významným dodávateľom valcovacích zariadení technológií a riadiacich systémov, ako napr. Danieli, ConverTeam, Acos Vilarés, Vítkovice Heavy Machinery, ... Rieši technické problémy a technologické inovácie na teplých a studených valcovacích tratiach ako napr. ArcelorMittal Ostrava, Severstal Čerepovec, ArcelorMittal Vanderbijlpark, CSN Voltaredunta, Třinecké železářny Třinec, ...

Hlavné aktivity sú riešenie technických problémov, spracovanie štúdií, konzultačná a poradenská činnosť v oblasti technológií valcovania za tepla aj za studena. Spoločnosť taktiež vyvíja nové, optimalizuje a upravuje súčasné softwary modulov riadiacich systémov valcovacích tratí a vyvíja špeciálne programy určené pre offline simulácie procesov valcovania a ochladzovania. Taktiež sa venuje počítačovému modelovaniu procesov tvárnenia a tepelného spracovania.

## 3. Použité technológie

### 3.1 ContourLib

ContourLib je grafická knižnica vytvorená firmou ITA spol. s.r.o. v čase, keď neboli ešte dostupné žiadne grafické knižnice pre prácu s grafmi a modelmi. Pracuje z väčšej časti s grafickou knižnicou OpenGL, ale niektoré časti sú napísané aj v GDI+. Z knižnice som používal časť, ktorá sa stará o vykresľovanie FEM modelov, ktoré sú používané pre simulácie

**Prehlásenie zástupcu spolupracujúcej právnickej alebo fyzickej osoby**

„Súhlasím so zverejnením tejto bakalárskej práce podľa požiadaviek čl. 26, odst. 9  
Studijného a zkušebného rádu pro studium v bakalárských programech VŠB-TU Ostrava.“

Dňa: 7.5.2014

.....  
podpis zástupcu



tepelnej charakteristiky materiálov, mechanického napätia rôznych zariadení a podobné. Knižnica je neustále rozširovaná a sú do nej pridávané nové komponenty pre ďalšie používanie. Napríklad komponent z opisovaného druhého projektu bude po dokončení taktiež pridaný do tejto grafickej knižnice.

## 3.2 MFC

Windows API je API vyvinuté firmou Microsoft pre operačný systém Microsoft Windows. Všetky programy pre systém Microsoft Windows musia komunikovať so systémom pomocou Windows API nezávisle na použitom programovacom jazyku. Toto API obsahuje základné funkcie, ako aj funkcie pre vytváranie užívateľského rozhrania, sieťové služby, funkcie pre prácu s registrom Windows a mnoho ďalšieho. MFC je knižnica, ktorá obalaňuje časti Windows API do ucelených C++ tried. Triedy spravujú väčšinu objektov vo Windows a preddefinujú napríklad okná alebo kontrolné panely. Knižnica MFC bola prvýkrát predstavená v roku 1992 pre 16 bitovú verziu OS. Knižnica MFC je neustále vyvíjaná a nové verzie sú vydávané zároveň s každou novou verziou vývojárskeho nástroja Microsoft Visual C++. Aktuálna verzia je 11.0 [2].

MFC je udalosťami riadené programovanie, teda program čaká na nejakú vyvolanú udalosť a podľa nej sa zachová. MFC používa architektúru Document/View – dokument/pohľad, čo je rozdelenie manipulácie s dátami na dve časti. Dokument uchováva uložené dáta, stará sa o možné ukladanie, načítanie a tlačenie dát. Pohľad sprostredkováva dáta uložené v dokumente na obrazovku a reaguje na interakcie s dátami, môže ich editovať, ukladať späť do dokumentu. Vo všeobecnosti jeden dokument obsluhuje viac pohľadov. MFC aplikácie môžu byť SDI – majú iba jeden dokument a viac pohľadov, alebo MDI – môžu mať viac dokumentov a ku každému dokumentu môže byť pripojených viac pohľadov [3].

Výhodou MFC je, že oveľa zjednodušuje programovanie oproti Windows API, pretože značné množstvo bežne používaných funkcionality má obalené do funkcií a makier, takže nie je potrebné viacnásobne písať rovnaký kód. Taktiež má výhodu, že oproti API postaveným na MFC má menšiu veľkosť výslednej aplikácie. MFC je veľmi rozšíreným nástrojom pre vývoj Windows aplikácií v C++. Pri projektoch firmy ITA sme používali vývojárske prostredie Microsoft Visual Studio 2008, takže sme pracovali s MFC vo verzii 9.0, ktorá pridávala nové vizuálne vlastnosti k predošlým verziám.

## 3.3 OpenGL

OpenGL je prostredie pre vývoj prenositeľných interaktívnych 2D a 3D aplikácií. Od svojho uvedenia v roku 1992 sa stalo priemyselne najviac používané a najpodporovanejšie 2D a 3D grafické API. Spolupracuje s grafickým procesorom prístroja, takže umožňuje hardvérovú akceleráciu renderovania. OpenGL má široké využitie od 3D animácií cez CAD po vizuálne simulácie. Špecifikáciu API spravuje nezávislé konzorcium OpenGL Architecture Review Board. API je úplne voľne dostupný multiplatformový štandard nezávislý na značke použiteľný vo väčšine programovacích jazykoch. Tento štandard sa neustále vyvíja a umožňuje prístup k novým inováciám hardwaru cez mechanizmus rozšírení. Taktiež zabezpečuje rovnaké vizuálne výsledky na akomkoľvek hardware schopného skompilovať API nezávisle na operačnom systéme. Silnou stránkou OpenGL knižnice je rozsiahla a neustále aktualizovaná dokumentácia k celému API. Aktuálna verzia OpenGL je 4.4. V projektoch sme pracovali však s verziou 1.1, keďže táto verzia je súčasťou operačného systému Windows. Použitím tejto staršej verzie je zabezpečené, že aplikácie budú spustiteľné na systéme Windows bez inštalácie ďalších dodatočných grafických knižníc [4].

### 3.4 GDI+

GDI+ je jednou z hlavných súčastí operačných systémov Windows. Bol uvedený v operačnom systéme Windows XP ako nástupca a rozšírenie knižnice GDI. Umožňuje aplikáciám používať grafiku a formátovaný text pre výstup na displej a tlačiarne. Aplikácie nepracujú priamo s grafickým adaptérom, ale pracuje s ovládačmi zariadení podľa požiadaviek aplikácií. GDI+ sa skladá z 3 základných častí: 2D vektorovej grafiky, spracovaní obrázkov a typografie. Pri použití knižnice GDI+ som pracoval s časťou 2D vektorovej grafiky, ktorá dokáže vykresľovať primitívne tvary ako čiary, krivky a geometrické útvary. API pracuje so súradnicami obrazových bodov. Tým, že knižnica nepracuje priamo s grafickým adaptérom má nevýhodu oproti OpenGL v rýchlosti renderovania zložitejších animácií. Pre jednoduchšie tvary je však oveľa jednoduchším nástrojom ako OpenGL, keďže OpenGL je nižšie programovacie rozhranie [5].

## 4. Zadané úlohy a postup pri riešení

### 4.1 Projekt 1

Prvým projektom, ktorý mi bol zadaný, bola drobná úprava na zobrazovaní valcov v programe RollFlex.

#### 4.1.1 Teória

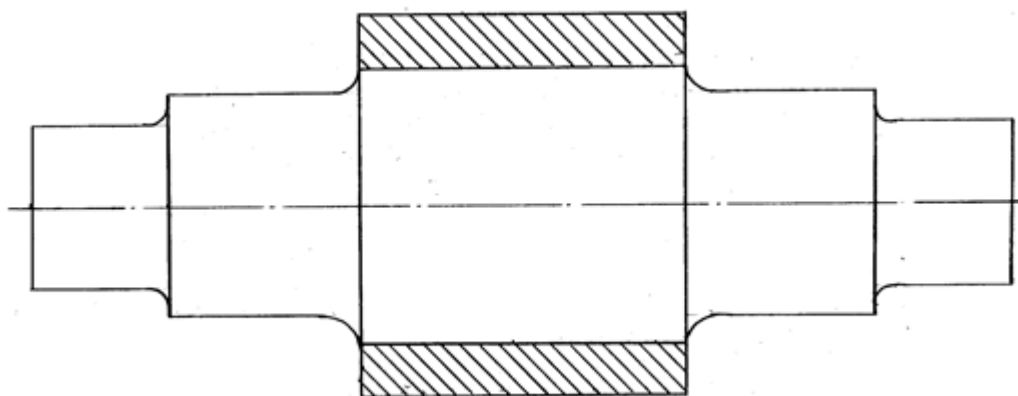
Do valcovacích stolíc sa používajú rôzne druhy valcov, ktoré majú odlišné spôsoby výroby. Valce môžu byť celé z jedného materiálu, alebo, hlavne pri väčších priemeroch, bývajú tieto valce z dvoch materiálov. Jeden materiál tvorí stred – jadro valca a druhý plášť. Pre jadro valca sa používa húževnatejší materiál, zatiaľ čo na plášť sa používa pevný, ale krehký materiál. Takáto kombinácia dosahuje požadované vlastnosti valcov – tvrdosť na povrchu, ale pevnosť vnútri.

Pre výrobu dvojvrstvových valcov sa používajú dve metódy:

- Metóda odstredivého liatia
- Nalisovanie plášťa na jadro

Metóda odstredivého liatia spočíva v tom, že roztavený materiál, z ktorého sa majú valce odlievať sa vlieva do rotujúcej formy. Do formy sa najskôr naleje materiál, ktorý má tvoriť plášť a ten sa odstredivou silou dostane na okraj formy. Následne sa do formy vleje materiál, ktorý má tvoriť jadro valca. Na hranici sa tieto dva materiály jemne premiešajú, teda sa vytvára plynulý prechod medzi jadrom a plášťom.

Druhá metóda spočíva v tom, že sa odleje zvlášť jadro a zvlášť plášť valca. Plášť sa následne nahreje na vysokú teplotu, čo spôsobí rozpínanie materiálu. Keď je plášť dostatočne rozťahnutý, nalisuje sa doňho jadro a nechá sa ochladiť, čím sa plášť znova zmrští a zabráni pohybu jadra cez neho. Obrázok 4.1 zobrazuje príklad pre valec vytvorený nalisovaním plášťa.



Obrázok 4.1 Prierez valca vytvoreného nalisovaním jadra do plášťa.

Program RollFlex vytvorený spoločnosťou ITA slúži na rôzne výpočty, optimalizácie a štúdie vykonávané na valcovacích stoliciach. Umožňuje napríklad výpočet tvaru valcovacej medzery sústavy valcov, výpočet pričného profilu plochých vývalkov, štúdium vplyvu rôznych akčných členov, tvorbu rôznych grafov závislých veličín a mnoho ďalších výpočtov. Súčasťou

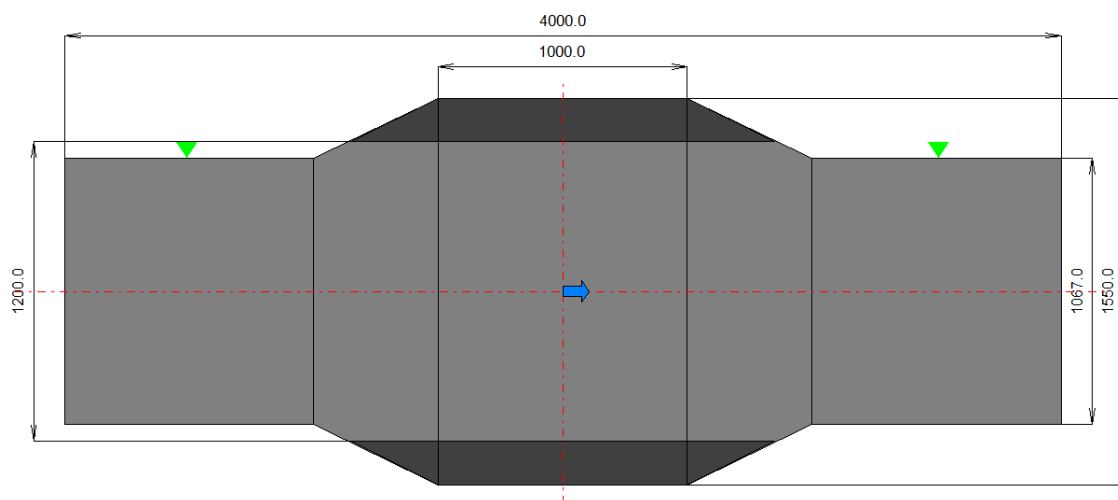
programu je aj vyobrazenie valcov v stoliciach, pre lepšiu vizualizáciu problému. Práve toto vyobrazenie bolo treba doplniť o možnosť zobrazovať aj dvojplášťové valce.

#### 4.1.2 Riešenie problému

Zobrazovanie valcov v programe bolo doteraz riešené jednofarebne. Kótovanie určovalo iba vonkajšie rozmery valca. Požiadavka bola, aby pri zobrazovaní v simulačnom programe bolo na prvý pohľad jednoznačné, že valec je vyrobený z dvoch rôznych materiálov, a takisto, aby bolo jasne viditeľné, akú časť tvorí jadro, a pokiaľ zasahuje plášť.

Všetky potrebné informácie pre vyobrazenie valca v signalizačnom programe sú obsiahnuté v triede `CRollGeometry`. Táto trieda sa stará aj o vykresľovanie samotného valca. Do tejto triedy boli pridané dva atribúty - `_coreDiameter` a `_coreDiameterFlag`. `_coreDiameterFlag` je typu `Boolean` a signalizuje, či je valec zložený z jadra a plášťa alebo len z jedného materiálu. `_coreDiameter` potom určuje priemer jadra. Do triedy boli ešte pridané dve metódy pre získanie týchto atribútov.

Pre vykresľovanie plášťa bola upravená len metóda `Draw`, ktorá sa starala aj o kompletne vykresľovanie celého valca. Pri inicializácii valca v programe je tento valec rozdelený na segmenty. Segmenty sa potom postupne vykresľujú v cykle `for`. Do vnútra cyklu som pridal overenie, či je nastavený parameter `_coreDiameterFlag` na `TRUE`, a teda či je valec zložený z dvoch častí. Ak program zistil, že áno, potom z rozmeru segmentu zistil, či je celý segment v medzi jadra, alebo presahuje až za priemer jadra – teda zasahuje do segmentu aj plášť. Ak je segment zložený z jadra a plášťa, program najskôr vykreslil segment, tak, ako keby bol z jedného materiálu, a potom vypočítal z priemeru a rozmeru segmentu body, ktoré tvoria hranicu medzi jadrom a plášťom. K týmto bodom sa pridali ešte body, ktoré tvoria kraj segmentu, a tento priestor medzi bodmi sa vykreslil tmavšou farbou. Po vykreslení všetkých segmentov je jednoznačné, ktorú časť tvorí jadro, a ktorú plášť. Metóda `Draw` taktiež vykresľuje kótovanie celého valca. K časti pre kótovanie som taktiež pridal overenie, či je valec zložený z jadra a plášťa, a ak áno, priemer tohto jadra sa tiež okótuje tak, aby sa neprekrýval s ostatnými kótami (vid' Obrázok 4.2).



Obrázok 4.2 Zobrazenie valca s jadrom a plášťom



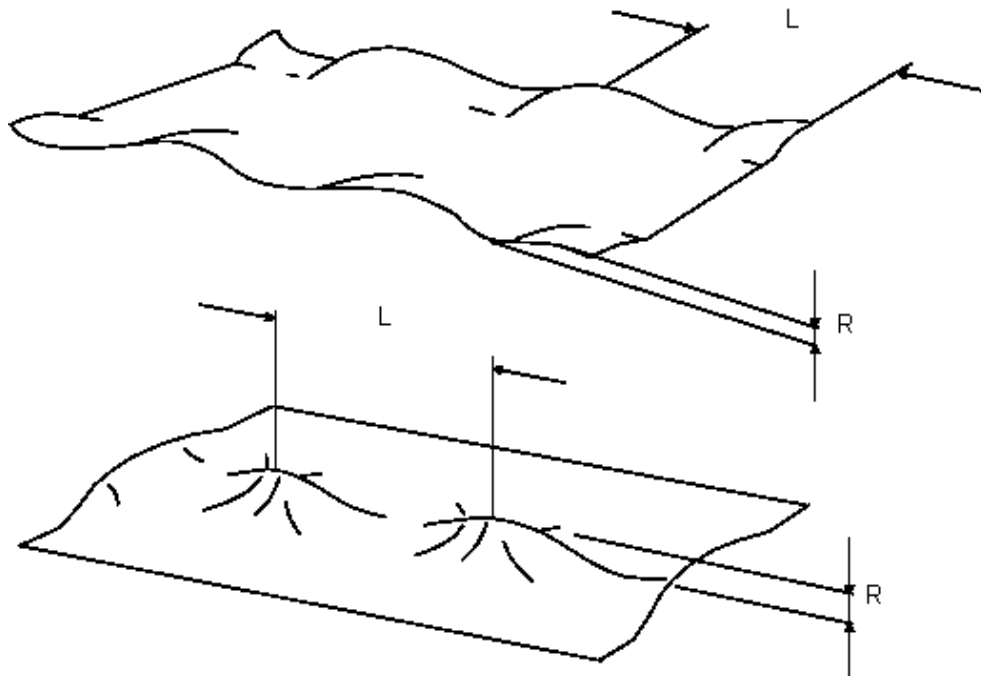
## 4.2 Projekt 2

Druhým mojim projektom bolo vytvorenie komponenty, ktorá dokázala vizualizovať dáta získané z novo nainštalovaného zariadenia, ktoré meralo rovinnosť oceľových pásov pri valcovaní za tepla, tak, aby boli čo najviac užívateľsky prívetivé. V tomto projekte som pracoval s grafickou knižnicou ContourLib, vytvorenou firmou ITA. Aby som sa bližšie oboznámil s knižnicou, bolo mi poskytnuté demo so zdrojovými kódmi, ktoré obsahovalo hlavné možnosti využitia tejto knižnice a príklady. Žiaľ, nebola k nej dostupná žiadna dokumentácia, takže som istý čas venoval štúdiu samotnej knižnice. V prípade nejasností som sa obrátil na pracovníkov vo firme, ktorí mi knižnicu bližšie predstavili.

### 4.2.1 Teória

Počas procesu valcovania oceľových pásov za tepla dochádza k rôznym nežiaducim deformáciám týchto pásov. Jednou z týchto deformácií je vlnitosť – porucha vlastnosti zvanej rovinnosť pásov. Vlnitosť môže byť dvojaká – stredové zakrivenia alebo okrajové vlny (vid' Obrázok 4.3). Táto vlnitosť je spôsobená pozdĺžnym pnúťím v pásoch. Toto pnutie má pôvod v zmenách profilu v jednotlivých valcovacích krokoch.

Na valcovni vo firme ArcelorMittal Ostrava bolo pridané nové zariadenie, ktoré meria túto rovinnosť. Zariadenie je umiestnené kolmo na pás a meria určitý počet bodov na šírke pásu a ako pás prechádza zariadením, postupne zmeria pás po celej svojej dĺžke. Tieto pásy môžu mať aj niekoľko kilometrov.



Obrázok 4.3 Vlnitosť pásov – hore okrajové vlny, dole stredové zakrivenia

Zozbierané údaje z prístroja bolo potrebné nejako vizuálne interpretovať, a k tomuto účelu som dostal za úlohu vytvoriť komponent, ktorý dáta vedel užívateľsky prívetivo vykresliť. V dobe vytvárania komponentu však neboli firme predané ani žiadne testovacie dáta, takže sme

sa dohodli, že zatiaľ budú vstup reprezentovať len náhodné dáta, ktoré by mohli reálne interpretovať dáta z prístroja. Hlavným dôvodom vytvárania komponentu bolo to, že oceľový pás môže mať aj viac ako kilometer, takže je nemožné ho zobraziť celý na obrazovku tak, aby neboli drobné výchylky skreslené a často vôbec neviditeľné. Preto bolo treba vytvoriť hrubý pohľad na celý pás, ktorý je značne zmenšený, a vytvoriť nejaký nástroj, ktorým by sa dal pás rozumne prehliadať v menších častiach, tak, aby bolo zreteľne viditeľné zvlnenie pásu v detailnejšom pohľade.

Grafická trieda `ContourLib` obsahuje komponent zvaný `FEMModel`. FEM je vo všeobecnosti metóda slúžiaca k simulácii priebehu napätia, deformácie, prúdenia tepla a pod. na vytvorenom fyzikálnom modeli. `FEMModel` v `ContourLib` obsahuje telá typu `CBody`. Hlavné časti potrebné pre vykreslenie tela `CBody` sú polia uzlov, elementov a kontúr. Uzly sú jednoducho body v 2D FEM modeli, ktoré obsahujú  $x$  a  $y$  súradnice bodu a svoj unikátny index. Elementy obsahujú hrany, ktoré spájajú jednotlivé uzly. Hrany identifikujú tieto uzly pomocou unikátnych indexov. Elementy sú hlavnou, vykresľovanou, časťou tela. V rámci jedného tela musia elementy na seba nadväzovať – telo nemôže byť rozdelené na dve úplne izolované časti. Ak si predstavíme elementy ako neorientovaný graf, tento graf musí byť spojený – teda existuje cesta v grafe z každého existujúceho uzla do každého iného uzla. Posledná časť – kontúry – sú `double` hodnoty každého uzla. Podľa kontúr grafická knižnica určuje, akú farbu má každý uzol a takisto plynulé farebné prechody medzi uzlami v rámci segmentov. Počet kontúr musí byť taký istý ako počet uzlov tela. Kontúry si potom knižnica automaticky priradí k indexom uzlov. Kontúry priradzuje k uzlom tak, že index kontúry v poli sa zhoduje s unikátnym indexom uzla. Tento komponent bol naprogramovaný pomocou MFC.

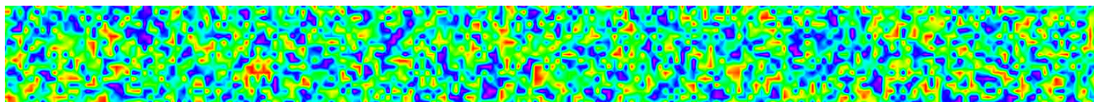
#### 4.2.2 Riešenie problému

Keďže v dobe písania tejto práce neboli ešte dodané reálne dáta získané zo zariadenia pre meranie rovinnosti, pre reprezentáciu nameraných dát som vytvoril jednoduché 2D pole, ktoré obsahovalo prvky typu `double`. Toto pole je uložené v dokumente MFC štruktúry a v jeho konštruktoze je nastavená veľkosť poľa. Pri vytvorení nového dokumentu sa alokuje 2D pole a naplní sa náhodnými hodnotami reprezentujúcimi vlnitosť pásu. Hlavná časť projektu – teda celý pás – je potomok MFC triedy `CWnd` a manažér, ktorý sa bude starať o vykresľovanie výberu z pásu. Odkaz na manažéra je uložený v triede `CWnd`, ktorá mu automaticky posiela oznámenia o všetkých udalostiach.

Potomok `CWnd` triedy si uchováva referenciu na dokument MFC štruktúry a z neho dokáže získať pole reprezentujúce pás. V tomto potomkovi sa vytvorí zo získaného 2D poľa `FEMModel`. Keďže pás je nutné kvôli jeho rozmerom vykresliť tak, že pre dĺžku sa použije oveľa menšia mierka ako pre šírku, bola v triede vytvorená premenná, ktorá určovala, v akom pomere sú tieto dve mierky. 2D pole sa teda postupne prejde dvoma cyklami `for` a vytvorí sa z neho 2 polia `FEMModelu`:

- Pole uzlov, ktorých súradnice sa vytvoria z indexov 2D poľa s tým, že  $y$  súradnica sa vynásobí pomerom mierok pre dĺžku a šírku
- Pole kontúr, ktorého hodnoty sú hodnoty konkrétnych buniek 2D poľa náležiacim indexu aktuálneho uzlu

Z vytvorených uzlov sa následne poskladajú elementy, ktoré sa pridajú do tela `CBody` a toto telo spolu s kontúrami sa zapíše do `FEMModelu`. Po vytvorení `FEMModelu` sa predá tento model manažérovi spolu s informáciou o veľkosti 2D poľa a pomere medzi mierkami. Keď sa do manažéra zapíše `FEMModel`, knižnica `ContourLib` sa postará o jeho vykreslenie do okna programu (vid' Obrázok 4.4). Následne sa zavolá vykresľovanie nástroja pre výber časti pásu.



Obrázok 4.4 Zobrazenie pásu s náhodnými hodnotami zakrivenia

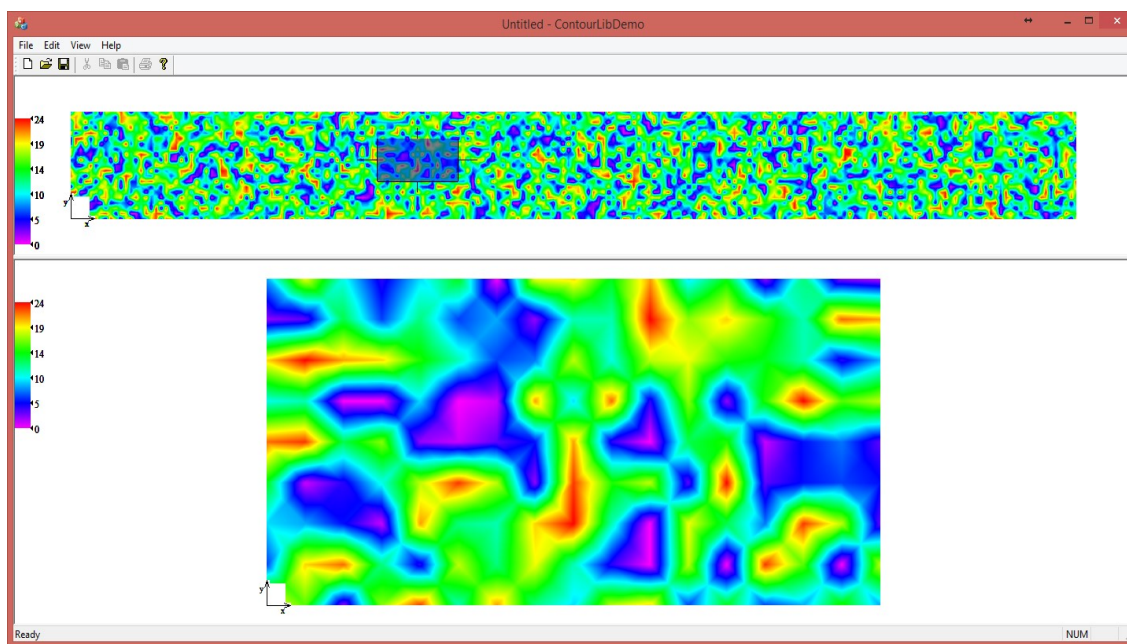
Nástroj pre výber časti som realizoval ako priesvitný obdĺžnik s krížom uprostred, ktorý sa dal rôzne pohybovať a posúvať po celom pásu. K vykresleniu som použil API OpenGL. V manažérovi som vytvoril metódu `OnAfterDrawScene`, ktorú volal manažér po vykreslení samotného `FEMModelu`. V tejto metóde sa získajú rozmery tela `FEMModelu` a táto veľkosť sa rozdelí podľa veľkosti poľa, ktoré reprezentuje pás, čím sa zistí výška a šírka jedného segmentu tela `FEMModelu` v OpenGL súradniciach. Aktuálna pozícia nástroja pre výber je teda vyjadrená v počte segmentov z ľavého okraja nástroja po ľavý okraj tela a z horného okraja nástroja po horný okraj tela. Taktiež rozmer nástroja je vyjadrený množstvom segmentov, ktoré zaberá na šírku aj výšku. Pri vykresľovaní segmentu sa tieto „dlaždicové“ súradnice prepočítajú späť na OpenGL súradnice. Uchovávanie v „dlaždicových“ súradniciach je volené pre uľahčenie oznamovania vybranej časti – odovzdávajú sa samotné súradnice bez zmeny – a kvôli tomu, aby vykresľovaný obdĺžnik vždy presne určoval aj pozíciu v segmentoch, keďže pri uchovávaní v OpenGL súradniciach by mohlo dôjsť k miernym nepresnostiam, kedy by vykreslený obdĺžnik nebol na okraji segmentu a programovo by sa javil presne od okraja.

V komponente boli taktiež upravené udalosti vyvolávané posunom kurzora po modeli a stlačenie ľavého tlačidla myši. Udalosť vyvolávaná pravým tlačidlom myši bola takisto prepísaná, ale ako prázdna funkcia, keďže vyvolávalo kontextovú ponuku, ktorá je v tomto prípade nežiaduca. Pohyb kurzora po modeli bol upravený tak, že keď je kurzor na obdĺžniku pre výber, zmení sa na ikonu posunu a stlačením ľavého tlačidla je ním možné posúvať po modeli. Ak je kurzor na okrajoch obdĺžnika, zmení sa na šípku pre zmenu veľkosti podľa toho, na ktorej hrane, resp. rohu sa aktuálne kurzor nachádza a po stlačení ľavého tlačidla je možné meniť veľkosť výberu. Všetky tieto manipulácie s nástrojom sú v metóde udalosti pre posun neustále kontrolované tak, aby nebolo možné posunúť nástroj, resp. zmeniť jeho veľkosť, tak, že by zasahoval mimo okraj tela, čo by vyvolalo výnimku.

Pre ďalšie komponenty, ktoré budú spracovávať údaje z výberu z pásu bolo vytvorené rozhranie `IContourSliderListener`. Toto rozhranie obsahuje virtuálnu metódu `OnDetailChanged`. Ku komponentu sa pripájali vo forme listenerov – poslucháčov. Komponent si uchovával zoznam všetkých pripojených listenerov a pri akejkoľvek zmene obdĺžnika pre výber vo všetkých pripojených listeneroch zavola metódu `OnDetailChanged`. V tejto metóde im predal pozíciu a veľkosť výberu v „dlaždicových“ súradniciach. Toto riešenie pomocou listenerov bolo volené, aby bolo možné zobrazit' rôzne pohľady na výber z pásu – napr. detail na vybraný úsek, keďže celý pás je značne zmenšený, skreslený po dĺžke

a v podstate neprehľadný, čo sa týka detailov, alebo prierez pásu v danom mieste pre lepšiu vizualizáciu zvlnenia pásu, keďže v pohľade z vrchu je táto vlnitosť zobrazovaná iba farebným odlíšením.

Pre demonštráciu a otestovanie funkčnosti som vytvoril v aplikácii dve zobrazenia CView oddelené splitterom z MFC knižnice. Jedno zobrazenie bol samotný vytváraný komponent, teda zobrazenie celého pásu s možnosťou výberu jeho úseku. Druhé zobrazenie bolo vytvorené pre demonštráciu funkčnosti, a zobrazovalo približený výrez z pásu v reálnych pomeroch – teda s rovnakou mierkou na výšku aj šírku. Toto zobrazenie vykresľuje a vytvára taktiež manažér, ale pôvodná, teda rodičovská trieda vytvoreného komponentu. Zobrazenie si uchováva odkaz na pole reprezentujúce pás z dokumentu. Pri vyvolaní funkcie OnDetailChanged vytvorí telo FEMModelu s rovnakým počtom segmentov, aký bol predaný vo funkcii OnDetailChanged ako šírka a výška výberu. Kontúry potom naplní z poľa reprezentujúceho pás s tým, že nezačne naplňať od indexu 0;0, ale od indexu x;y predaných vo funkcii. Takto vznikne FEMModel veľkosti výberu v reálnej mierke a hodnotami súhlasiacimi s hodnotami vo výbere (vid' Obrázok 4.5).



Obrázok 4.5 Zobrazenie výrezu v druhej komponente

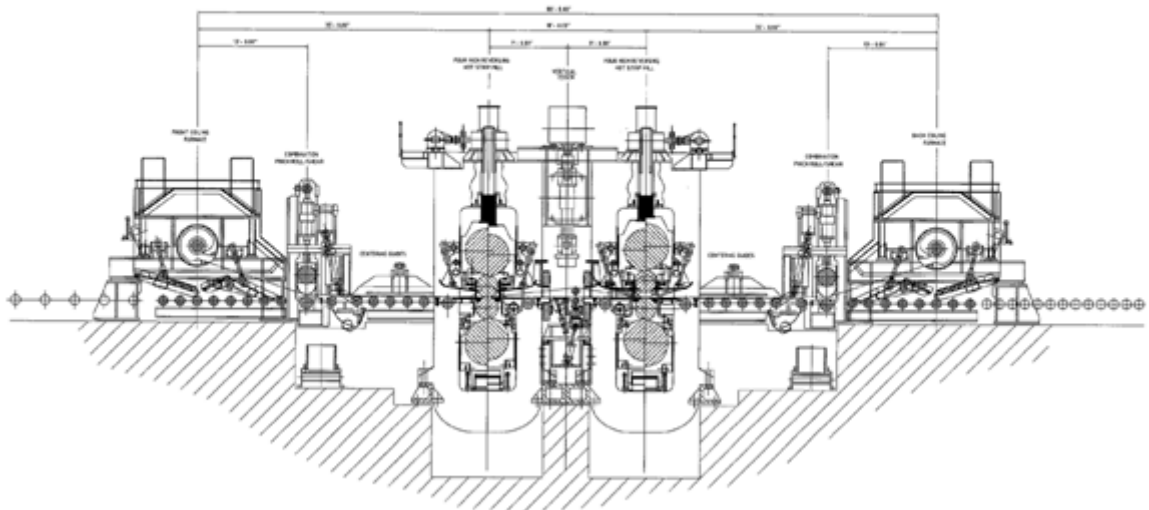
### 4.3 Projekt 3

Tretím mojim projektom bolo vytvoriť jednoduchú schému tzv. Hot strip millu, pre spoločnosť ArcelorMittal.

#### 4.3.1 Valcovňa za tepla

Súčasťou valcovacieho procesu oceľových pásov sú valcovacie stolice. Vo firme ArcelorMittal Ostrava je použitý takzvaný Twin steckel mill, čo sú dve kvatro stolice – každá stolica má dva operné, väčšie valce a medzi nimi dva pracovné valce. Z pece prichádza na valcovacie stolice hrubý kovový pás, ktorý prechádza valcovacími stolicami počas viacerých prechodov tam a späť a je na ňom prevádzaný tzv. hrubý úber – uberá sa s pásu s dôrazom na to,

aby bola zakaždým ubratá čo najväčšia možná časť pásu. Úberom z hrúbky pásu sa materiál presúva na šírku, takže s každým úberom sa pás značne predlžuje. Pred aj za stolicami sú umiestnené pecné navíjačky, pretože po niekoľkých prechodoch stolicami sa už pás kvôli svojej dĺžke nezmesť na vodiace pásy, takže sa navíjajú do navíjačiek. Ďalším významom navíjačiek je rovnomerné udržiavanie a rozkladanie tepla v páse. Správne, rovnomerne rozložené teplo v páse je dôležité kvôli výsledným mechanickým vlastnostiam pásu, prípadne plechov z neho natrihaných. Obrázok 4.6 zobrazuje takéto usporiadanie valcovacích stolíc s pecnými navíjačkami. Z pohľadu schémy prichádza pás na valcovacie stolice z ľavej strany – od pece. Následne prejde stolicami a keďže nie je ešte moc predĺžený, pokračuje na vodiaci pás. Z vodiaceho pásu ide späť na stolice a na vodiaci pás od pece. V následnom prechode je už pás príliš dlhý, aby sa zmestil na vodiaci pás, takže sa začne navíjať do pravej navíjačky. Z pravej navíjačky ide späť cez stolicu do ľavej a potom znova späť do pravej. Tento proces sa pár krát opakuje, až je dosiahnutá požadovaná hrúbka pásu a pás pokračuje po dopravníku na ďalšie spracovanie smerom doprava.



Obrázok 4.6 Twin steckel stolice s pecnými navíjačkami pred aj po stolicach

Schéma týchto stolíc je určená pre riadiaci program valcovne, aby bolo na prvý pohľad jasné, kde sa momentálne nachádza valcovaný pás. Taktiež bola požiadavka, aby schéma dokázala zobrazovať, kde bude pás smerovať po prechode pecou, či bude pokračovať na vodiaci pás, alebo bude navitý do pecnej navíjačky. Za stolicou je umiestnené meracie zariadenie, ktoré bolo treba taktiež zakomponovať do schémy a zobraziť na ňom, či je momentálne merací prístroj v činnosti.

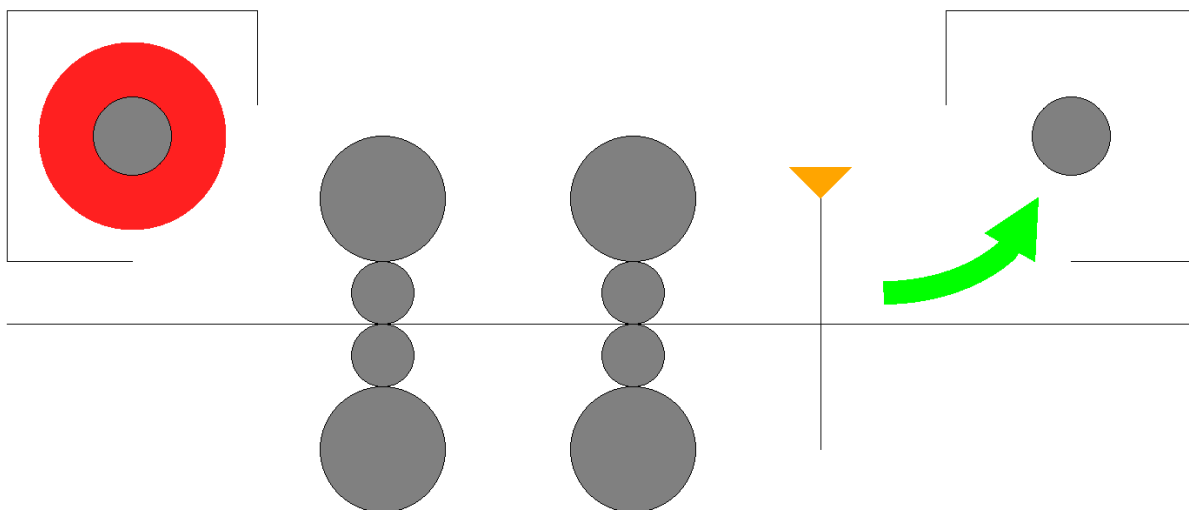
#### 4.3.2 Riešenie problému

Dohodli sme sa, že schému budeme realizovať ako jednu funkciu, ktorá bude prijímať argumenty reprezentujúce polohu horúceho pásu kovu vo valcovacej stolici. Vytvoril som teda funkciu `DrawSteckel(bool pLeft, bool pReel, bool pToReel, bool pMeasuring, CRect pRect, CDC* pDC)`.

## Argumenty:

- `pLeft` signalizuje, či je pás na ľavej alebo pravej strane stolice (ľavá strana je vstup do valcovacej stolice, pravá výstup)
- `pReel` signalizuje, či je pás na navíjačke, alebo na stole
- `-pToReel` signalizuje, či pás po skončení valcovania smeruje do navíjačky, alebo na stôl
- `-pMeasuring` signalizuje, či je merač zapnutý a meria, alebo vypnutý
- `-pRect` udáva obdĺžnik v obrazkových bodoch, do ktorého bude schéma vykresľovaná
- `-*pDC` je referencia na *handler* kresliacej plochy

Pre vykreslenie celej schémy som použil grafickú knižnicu *GDI+*. Celý priestor pre vykresľovanie som si rozdelil na mriežku o veľkosti 41 x 19. Najskôr som si zistil, aká je veľkosť obdĺžnika `pRect`, a vypočítal veľkosť dlaždíc a odsadenie od vrchného, resp. ľavého okraja priestoru pre vykreslenie tak, aby bola vždy vyplnená čo najväčšia časť možného priestoru, ale zároveň žiadna časť schémy nepresahovala mimo okraj priestoru. Aby nebola schéma deformovaná pri akejkoľvek veľkosti a pomere strán priestoru pre vykreslenie schémy, dlaždice boli štvorcového tvaru a všetky súradnice vykresľovaných segmentov schémy boli násobky tejto veľkosti dlaždíc, nie konkrétne obrazkové body. Schéma bola vykreslená ako veľmi zjednodušená verzia dvoch stolíc a navíjačiek. Žeravý pás je vykreslený červenou farbou a smer, kam pôjde po prechode stolicami je vyobrazený zelenou šípkou. Obrázok 4.7 zobrazuje príklad výslednej schémy. V tomto konkrétnom prípade je horúci pás navitý na ľavú navíjačku, po ďalšom prechode stolicou bude smerovať do pravej navíjačky. Takisto je pre demonštráciu merač zapnutý – je vyplnený oranžovou farbou. Prekreslenie tejto schémy sa vykonáva vždy pri zavolaní funkcie `DrawSteckel`.



Obrázok 4.7 Schéma valcovacej stolice

## 5. Teoretické a praktické poznatky

Pri vykonávaní praxe pre firmu ITA spol. s.r.o. som uplatnil niektoré znalosti nadobudnuté z predmetov absolvovaných počas štúdia. Moja práca spočívala v programovaní v jazyku C++, takže uplatnenie našli v podstate celé moje prvé dva roky na vysokej škole. V prvom ročníku bolo naše prvé stretnutie s programovaním práve prostredníctvom jazyka C++. V druhom ročníku bolo štúdium jazyka C++ zamerané hlavne na objektové programovanie, spolu s prechodom na jazyk C#, ktorý je už od podstaty objektovo orientovaný. Keďže sa jednalo aj o prácu s počítačovou grafikou, ktorá je založená na matematike, uplatnil som aj znalosti z predmetov venovaných sa práve nej. Počas štúdia sme taktiež pracovali s grafickými možnosťami CUDA architektúry a s vývojárskymi nástrojmi pre túto technológiu, takže som mal osvojené základy programovania jednoduchej grafiky, ktoré mi pomohli pri štúdiu technológie OpenGL. Túto technológiu som použil ako pri aplikácii pre vstupný pohovor, tak aj pri realizácií samotných komponentov, ktoré boli súčasťou praxe.

Počas praxe som však objavil aj veľa nedostatkov, hlavne s technológiou MFC, s ktorou som nemal predtým žiadne skúsenosti a s vlastnou knižnicou ContourLib, ktorú vytvorila firma ITA. Obe tieto knižnice boli zložitejšie, takže som im venoval väčšiu pozornosť. K technológii MFC boli verejne dostupné manuály a návody, prípadné nezrovnalosti mi boli objasnené pri návšteve firmy. S knižnicou ContourLib to bolo mierne zložitejšie, keďže to je súkromné vlastníctvo firmy bez akejkoľvek verejnej dokumentácie, takže pri jej štúdiu som sa musel spoľahnúť na pomoc spolupracovníkov vo firme a na poskytnuté demo, ktoré demonštrovalo základné možnosti tejto grafickej knižnice. Novinkou, a veľmi cennou skúsenosťou, bola pre mňa taktiež práca v tíme, a vytváranie izolovanej časti programu, ktorá sa dá použiť v rámci väčšieho celku.

## 6. Záver

Rozhodnutie vykonať odbornú prax namiesto klasickej bakalárskej práce splnilo moje očakávania. Prínosom bolo reálne riešenie zadaného problému s konkrétnymi požiadavkami a práca v tíme, s čím som sa v škole nestretol. Ocenil som zužitkovanie teoretických znalostí nadobudnutých počas štúdia. Taktiež som sa naučil mnoho nových vecí, ako napríklad technológiu MFC, o ktorej som predtým nevedel, a zistil som, že je značne rozšírená. Získal som možnosť ďalej rozširovať základné znalosti o tejto technológii, čo mi, dúfam, pomôže v hľadaní budúceho zamestnania. Som veľmi rád, že som mohol prax vykonať vo firme ITA spol. s r.o., kde mi bolo umožnené získať nový pohľad na programovanie, ktoré som tu využil hlavne pre simulačné a grafické účely. Zistil som však, že mám aj značné nedostatky v znalostiach, a že aj po skončení štúdia na vysokej škole bude stále veľa priestoru na zlepšovanie sa v mnohých smeroch. Hlavným pozitívom však ostáva, že som získal cennú skúsenosť s programovaním formou reálneho využitia v skutočných projektoch, nie len ako forma výučby.



## Použitá literatura

- [1] ITA-tech s.r.o. *Společnost ITA* [online]. 2007 [cit. 2014-04-25]. Dostupné z: <http://www.ita-tech.cz/>
- [2] MFC Desktop Applications. *MSDN-the microsoft developer network* [online]. 2013 [cit. 2014-04-22]. Dostupné z: <http://msdn.microsoft.com/en-us/library/d06h2x6e.aspx>
- [3] ECKERT, Richard. MFC WINDOWS PROGRAMMING: The Microsoft Foundation Class (MFC) Library. *Binghamton University - Watson: Computer Science* [online]. 2000 [cit. 2014-04-22]. Dostupné z: <http://www.cs.binghamton.edu/~reckert/360/class14.htm>
- [4] OpenGL Overview. *OpenGL - The Industry Standard for High Performance Graphics* [online]. [cit. 2014-04-22]. Dostupné z: <http://www.opengl.org/about/>
- [5] GDI+ (Windows). *Windows Dev Center Home* [online]. 2012 [cit. 2014-04-23]. Dostupné z: <http://msdn.microsoft.com/en-us/library/windows/desktop/ms533798.aspx>